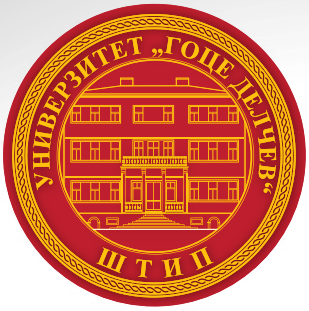

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ” - ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ISSN:1857-8691

ГОДИШЕН ЗБОРНИК
2013
YEARBOOK
2013

ГОДИНА 2 **VOLUME II**

GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ГОДИШЕН ЗБОРНИК
2013
YEARBOOK
2013

ГОДИНА 2 МАРТ, 2014 VOLUME II

GOCE DELCEV UNIVERSITY – STIP
FACULTY OF COMPUTER SCIENCE

ГОДИШЕН ЗБОРНИК
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
YEARBOOK
FACULTY OF COMPUTER SCIENCE

За издавачот:
Проф д-р Владо Гичев

Редакција и администрација
Универзитет „Гоце Делчев“-Штип
Факултет за информатика
ул. „Крсте Мисирков“ 10-А
п. фах 201, 2000 Штип
Р. Македонија

Address of the editorial office
Goce Delcev University – Stip
Faculty of Computer Science
Krste Misirkov 10-A
PO box 201, 2000 Štip,
R. of Macedonia

СОДРЖИНА
CONTENT

S-BOXES – PARAMETERS, CHARACTERISTICS AND CLASSIFICATIONS

Dusan Bikov[1], Stefka Bouyuklieva[2] and Aleksandra Stojanova[3]

[1] "Goce Delcev" University – Stip
(dusan.bikov, aleksandra.stojanova)@ugd.edu.mk
[2] "St.Cyril and St.Methoduis University" of Veliko Tarnovo
(stefka_iliya)@yahoo.com

Abstract. S-Boxes are key building blocks in the design of the block ciphers. They are basically used to hide the relationship between the plain text and the cipher text.
In this paper we study the parameters of Boolean functions and S-boxes, which are important in the design of good cryptosystems. We give a brief overview of the selection criteria on S-boxes, which can be resistant to different type of cryptanalytic attacks. For this goal optimality of S-box is defined. We present different variants for classification of S-boxes and give some examples. Also we list the results of our computer calculations for the parameters of Boolean functions and S-boxes that are essential in the cryptographic research. Finally, we give general framework of the direction in which our study is focused.

Keywords: Boolean function, Differential cryptanalysis, Linear cryptanalysis, Affine equivalence.

1 Introduction

In his paper "Communication Theory of Secrecy Systems" from 1949, Claude Shannon introduced some design principles for ciphers [1]. He proposed *confusion* and *diffusion* in the encryption algorithms. Cryptosystems are still designed according to these principles. The key elements in almost all block ciphers are the substitution boxes (S-boxes), which are used to ensure the confusion [1] of the information.

S-boxes form the non-linear part of a block cipher and therefore they are very important for the security of these ciphers. S-boxes have to be chosen carefully, in order to make the cipher resistant against different attacks. Thus, the generation and classification of small S-boxes with good linear and differential properties is very helpful. The S-box is a function S with values that are bit strings, or

$$S: F_2^n \to F_2^m$$

In many cases it is represented by a table. For any vector $v \in F_2^m$ a component function $S_b: F_2^n \to F_2$ is defined by $S_b(x) = b \cdot S(x)$. As S_b are Boolean functions, some their parameters and properties are very important in the design of S-boxes.

We take into account the following parameters of an S-box:

- Difference distribution table.
- Differential Uniformity (Diff(S) or Δ(S)).
- Linear approximation table.
- Linearity, linear probability and linear probability bias.
- Branch number.

We give some examples for 4×4, and greater S-boxes and present some classification results in the 4×4 case.

We can generate good S-boxes with two primary ways: (1) picking a random large S-box or (2) generating small S-boxes with good linear and differential properties. The main drawback of picking a random large S-box, is that these S-boxes are much more inefficient to implement, especially in hardware [2].

It is difficult to find an optimal S-box, because of a huge number of permutations for small values of n-bits S-box. For example, the number of 4-bit permutations is still huge: roughly 2^{44}. Because of this, after exhaustively checking all, finding good S-box, is no option. Resistance of S-box against most attacks remains unchanged, when invertible affine transformation before and after the S-box is applied. This reduction allows us to check all optimal S-boxes thoroughly, with consideration to the other criteria, such as algebraic degree.

1.1 Overview of this paper

In section 2, we give s-box properties notation. In section 3, we find parameters for example 4-bit S-box, and we show results for testing S-boxes with different size. In section 4, we define optimal criteria. In section 5, we suggest further ideas to be investigated. We conclude in section 6.

2. S-Box Properties - Notation

Let $F_2 = \{0,1\}$ be a finite field with two elements and F_2^n be the n-dimensional vector space over F_2. A Boolean function in *n* variables is a function $f: F_2^n \to F_2$ which maps any binary vector of length *n* (n-tuple or *n* bit input) to 0 or 1. A common way of representing a Boolean function is by supplying a list of output values for each n-bit input vector, called the truth table of the function. Actually this is a vector consisting of all the outputs which we obtain for the lexicographically ordered inputs:

$$f \mapsto v_f = (v_0, v_1, \dots, v_{2^n-1}) \in F_2^{2^n},$$

where $v_i = f(\bar{\imath}), \bar{\imath}$ is the binary representation of the integer *i*. The number of all Boolean functions in n variables is 2^{2^n}.

Every Boolean function can be written as a polynomial:

$$f(x_1, x_2, \dots, x_n) = \sum_u c_u x^u, \qquad (1)$$

where $c_u \in F_2, x^u = x_1^{u_1} x_2^{u_2} \dots x_n^{u_n}, u = (u_1, u_2, \dots, u_n) \in F_2^n$. This presentation is called Algebraic Normal Form (ANF) of the function. The degree of the polynomial (1) is the algebraic degree of *f* (deg f). Obviously, the maximum degree of a Boolean function in *n* variables is *n*.

A Boolean function with algebraic degree at most 1 is called affine, so the function *f* is affine Boolean function if

$$f(x_1, \dots, x_n) = a_0 + a_1 x_1 + \dots + a_n x_n = a \cdot x + a_0,$$

where $a \in F_2^n, a_0 \in F_2$. If $a_0 = 0$, the affine function is called linear.

Nonlinearity nl (f) of the Boolean function f is the minimal Hamming distance from f to the affine functions:

$$nl(f) = \min\{d(f, g) \mid g - affine\ function\}$$

The nonlinearity is at most $2^{n-1} - 2^{n/2-1}$ [7]. For cryptographic Boolean functions, *nl (f)* must be close to this maximum to prevent the system from attacks by linear approximations, correlation attacks, fast correlation attacks etc. [8].

A Boolean function *f* on F_2^n is also uniquely determined by its Walsh transform. The Walsh transform f^W of f is an integer valued function defined by

$$f^W(a) = \sum_{x \in F_2^n} (-1)^{f(x)+\langle a,x \rangle} = 2^n - 2d_H(f, f_a)$$

where $\langle a, x \rangle$ is scalar product.

Linearity *Lin (f)* of the Boolean function *f* is defined by using Walsh transform

$$Lin(f) = \max_{a \in F_2^n} |f^W(a)| \geq 2^{n/2}.$$

Linearity and nonlinearity of a Boolean function are connected by the equality

$$nl(f) = 2^{n-1} - \frac{1}{2} Lin(f).$$

From mathematical point of view S-box (or vectorial Boolean function) is a function S, with values that are bit string, or mapping of n bits to m bits

$$S: F_2^n \to F_2^m.$$

For any vector $b = (b_1, b_2, \dots b_m) \in F_2^m$ we consider the corresponding component function $S_b: F_2^n \to F_2$ defined by

$$S_b(x) = \langle b, S(x) \rangle = b_1 S_1(x) + \dots + b_m S_m(x).$$

2.1 Linear cryptanalysis (LC)

Linearity is a measure for resistance against linear cryptanalysis [3]. We define linearity of S as

$$Lin(S) = \max_{a \in F_2^n, b \in F_2^m, b \neq 0} |S_b^W(a)| = \max_{b \in F_2^m, b \neq 0} Lin(S_b)$$

In the theory of block ciphers related to linear cryptanalysis, the linear approximation table is studied. The linear approximation table is a $2^n \times 2^m$ table whose entries are defined as

$$L_{a,b} = \#\{x \in F_2^n : \langle b, S(x) \rangle = \langle a, x \rangle\} = 2^n - d_H(S_b, f_a)$$

The probability of a linear approximation of a linear combination of output bits S_b by a linear combination of input bits we define as

$$p_{a,b} = \frac{1}{2^n} L_{a,b}.$$

Linear probability bias ε is a correlation measure for this deviation from the probability ½ for which it is entirely uncorrelated:

$$\varepsilon_{a,b} = \left| p_{a,b} - \frac{1}{2} \right| \leq \left| \frac{Lin(S)}{2^{n+1}} \right|$$

The smaller is the linearity more resistant is the S-box against linear cryptanalysis. An open problem for given integers and m is to find nxm S-boxes with the smallest linearity.

2.2 Differential cryptanalysis (DC)

Differential cryptanalysis is proposed by Biham and Shamir [4], and is basically applied to block ciphers. This attack keeps up with the differences in the propagation during the encryption of the messages *m* and *m+δ* through the different rounds in a block cipher. Here a difference distribution table DDT is defined as

$$D_{a,b} = \#\{x \in F_2^n : S(x) \oplus S(x \oplus a) = b\}.$$

Similarly to the linear case, a differential probability is defined as

$$DP_{a,b} = \frac{1}{2^n} D_{a,b}.$$

To measure the resistance against differential cryptanalysis we take the highest possible value in DDT called differential uniformity

$$Diff(S) = max\{D_{a,b}, a \in F_2^n, a \neq 0, b \in F_2^m\}.$$

Diff(S) is related to the maximal probability that any fixed nonzero input difference causes any fixed output difference after applying the S-box.

2.3 Branch number

An important parameter describing the diffusion capabilities is the branch number. Branch number [5] is defined as

$$BN(S) = \min_{a,b \in F_2^n, a \neq b} (w_H(a \oplus b) + w_H(S(a) \oplus S(b)))$$

where w_H is the Hamming weight and S the S-box.

The branch number here depends on the position of the values in the difference distribution table. For bijective S-boxes BN ≥2. Branch number is related to the avalanche property [9] of the S-box and should be as greater as possible. In [6] differential branch number and linear branch number are defined.

3. Finding S-boxes parameters

Here we calculate some parameters of the 4-bit S-box G_3, which is one of the 16 different optimal S-boxes classified in [2]. We present G_3 by the following table:

Table 1. S-box G_3

a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S(a)	0	1	2	13	4	7	15	6	8	12	5	3	10	14	11	9

It can be represented also as a permutation, in this case this is the 16-tuple with values from the second row of Table 1: (0, 1, 2, 13, 4, 7, 15, 6, 8, 12, 5, 3, 10, 14, 11, 9), so

G_3: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) → (0, 1, 2, 13, 4, 7, 15, 6, 8, 12, 5, 3, 10, 14, 11, 9).

Replacing every number by its binary 4-bit string, we obtain

$$(0,1,2,13,4,7,15,6,8,12,5,3,10,14,11,9) \rightarrow \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

To calculate the linearity Lin(S) of S, we use the first order Reed-Muller code RM (1, 4). The set of all binary vectors (true tables) corresponding to the affine Boolean functions in n variables, coincides with the first order Reed-Muller code RM (1, n). It is a linear code of length 2^n, dimension n+1 and minimum distance 2^{n-1}.

RM (1, 4) has a generator matrix:

$$G(RM(1,4)) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

For the linearity of G_3 we have $nl(S_b) = d(S_b, RM(1,4))$

$$\Rightarrow Lin(S_b) = 2^4 - 2nl(S_b) = 16 - 2d(S_b, RM(1,4))$$

$$Lin(S) = \max Lin(S_b) = 8$$

For the 4-bit S-box G_3 we calculate Linear Approximation Table (see Table 2). To do that, we wrote the program S-box_LATv0.2 in C++ programming language. The results from the calculations are saved in a text file.

Table 2. LAT for G_3

a\x	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	4	0	4	4	8	-4	0	0	4	8	-4	-4	0	-4	0
0010	0	8	4	4	0	0	4	-4	0	-8	4	4	0	0	4	-4
0011	0	4	4	0	4	0	0	4	0	4	-4	8	-4	-8	0	4
0100	0	0	8	0	4	4	-4	4	4	-4	-4	-4	8	0	0	0
0101	0	4	0	-4	0	4	8	4	4	0	-4	0	-4	8	-4	0
0110	0	0	4	4	4	-4	8	0	-4	4	0	-8	0	0	4	4
0111	0	-4	-4	8	0	4	4	8	-4	0	0	4	4	0	0	-4
1000	0	0	0	0	-4	4	4	-4	8	8	0	0	4	-4	4	-4
1001	0	4	0	4	0	-4	0	-4	0	4	0	4	8	4	-8	4
1010	0	-8	4	4	4	-4	0	0	8	0	4	4	-4	4	0	0
1011	0	4	4	0	-8	-4	-4	8	0	4	4	0	0	4	4	0
1100	0	0	0	-8	8	0	0	0	-4	4	4	4	4	4	4	-4
1101	0	4	-8	4	4	0	-4	0	4	0	-4	0	0	4	8	4
1110	0	0	-4	-4	0	0	4	4	4	-4	8	0	4	-4	0	8
1111	0	-4	4	0	-4	8	0	-4	-4	0	0	4	0	4	4	8

Moreover, we wrote the program S-box_DDTv0.3 in C++ to calculate DDT (Difference Distribution Table) for the same 4-bit S-box G_3 (see Table 3). The results from the calculations are saved in a text file.

Table 3. DDT for G_3

a\x	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	-	2	2	2	4	-	2	-	-	2	-	-	-	-	-	2
0010	-	4	2	-	-	-	-	2	-	-	-	2	2	2	-	2
0011	-	-	2	4	-	2	-	-	2	2	-	2	-	2	-	-
0100	-	-	2	-	2	2	4	2	2	-	-	-	2	-	-	-
0101	-	-	2	-	2	2	4	2	2	-	-	-	2	-	-	-
0110	-	-	-	2	-	2	2	2	-	-	2	-	-	2	-	4
0111	-	2	-	-	-	2	2	2	-	4	-	2	-	-	2	-
1000	-	-	-	-	2	-	-	2	2	2	-	-		2	4	2
1001	-	2	-	-	-	-	2	-	2	2	2	-	2	4	-	-
1010	-	2	2	-	-	4	-	-	2	-	2	-	-	-	2	2
1011	-	2	-	2	2	2	-	-	-	-	-	-	4	2	2	-
1100	-	-	-	-	2	2	-	-	-	2	4	2	2	-	-	2
1101	-	-	-	2	-	-	2	-	2	-	-	-	4	2	2	2
1110	-	2	-	2	2	-	-	2	4	-	2	2	-	-	-	-
1111	-	-	2	2	-	-	-	4	-	2	2	-	2	-	2	-

Using the program S-box_DDTv0.3, we also calculated

$$Diff(S) = \max_{a \neq 0,\ b} D_{a,b} = 4;\ \ DP_{a,b} = \frac{D_{a,b}}{2^n} = 0, \frac{1}{4}\ or\ \frac{1}{8};\ \ BN = 2.$$

The mentioned program can be used to calculate the numbers in DDT, and the parameters, connected with this table, for bijective S-boxes with different lengths. We test the program for a resource computer consummation for S-boxes of different sizes (see Table 4). To generate the considered S-boxes, we use the program Generate S-box (written in C++).

Table 4. Calculations for DDT in different S-boxes (S-box_DDTv0.3)

Platform	**Intel(R) Core(TM)2 Duo CPU E8300 @2.83 GHz, 2 GB RAM (VS2013, C/C++)**		
S-box	Running Time:	RAM:	Size of the text file with the results:
4x4-bit	0.002 sec	…	3 KB
6x6-bit	0.015 sec	…	41 KB
8x8-bit	0.155 sec	1-2MB	771 KB
10x10-bit	3.619 sec	16 MB	14.5 MB
12x12-bit	108.185 sec	80-160 MB	260MB
13x13-bit	669.195 sec	400-500 MB	*132 MB
14x14-bit	12108.8 sec	550-1100 MB	*525 MB
Platform	**AMD Turion X2 Mobile TL-56 1.79 GHz, 2 GB RAM (VS2010, C/C++)**		
S-box	Running Time:	RAM:	Size save result text file from Calculation:
4x4-bit	0.002 sec	…	3 KB
6x6-bit	0.046 sec	…	41 KB
8x8-bit	1 sec	1-2 MB	771 KB
10x10-bit	24 sec	10-20 MB	14.5 MB
12x12-bit	758 sec	85-170 MB	260MB
13x13-bit	3920.91 sec	260 – 550MB	*132 MB
14x14-bit	…	…	…

* Without the values of S(x)⊕S (x⊕a), because the text file becomes very large with these values.

4. Optimal 4 Bit S-Boxes

A natural requirement for the S-boxes is their optimal resistance against linear and differential cryptanalyses. Unlike for higher dimensions the optimal values for Lin(S) and Diff(S) are known for the 4-bit S-boxes. More precisely, Lin(S) ≥ 8 and Diff(S) ≥ 4 (see [2]). More formally, as it is given in [2], the definition of an optimal 4-bit S-box is the following:

Definition 1. Let $S: F_2^4 \to F_2^4$ be an S-box. If S fulfills the following conditions we call S an optimal S-box.

1. S is a bijection.
2. Lin(S) = 8.
3. Diff(S) = 4.

When designing a block cipher it is important to know the set of S-boxes to choose from in order to get an optimal resistance against known attacks. Number of all permutations on F_2^n is 2^n and even for small dimensions n, it is crucial to reduce the number of S-boxes which have to be considered.

It is well known (see for example [8]) that the values of Diff(S) and Lin(S) remain unchanged if we apply affine transformations in the domain or co-domain of S. In particular if we take an optimal S-box in the above sense and transform it in an affine way, we get another optimal S-box. That's why we could find only representatives of the different equivalence classes. The definition for the affine equivalence is the following:

Definition 2. The S-boxes $S_1, S_2: F_2^n \to F_2^m$ are affine equivalent if

$$S_2(x) = B \cdot S_1(A \cdot x \oplus a) \oplus b$$

where A and B are invertible $n \times n$ and $m \times m$ matrices, respectively, $a \in F_2^n, b \in F_2^m$.

In [2] Leander and Poschmann proved that there are only 16 different optimal 4-bit S-boxes up to affine equivalence. In [9] Saarinen extends on this work by giving further properties of the optimal S-Box equivalence classes. He defines two S-Boxes to be cryptanalytically equivalent if they are isomorphic up to the permutation of input and output bits and a XOR of a constant in the input and output. In [5] the authors consider all invertible 4-bit S-boxes and search for most efficient S-box in each equivalence class.

5. Future Work

This research is focused on the calculation of the most important parameters of given S-boxes, generation of optimal S-boxes and classification of all (or only the optimal) S-boxes of given size. We are going to improve our technique in two ways:

- Optimization of the algorithms and search for new effective methods for computations. Our examples in this work are mainly for 4 × 4-bit S-boxes. The application of the used methods to calculate the parameter for large S-boxes is not feasible with a conventional computer and basic architecture.
- Parallel programming.
  Modern processors offer more advanced techniques, such as parallelism, pipelining and instruction set extensions. Using these features for S-box implementations can result in other tradeoffs which could be investigated. Selection of a technology for parallel programming combined with different optimizing methods can ensure promising results.

6. Conclusion

S-boxes form the nonlinear part in the block ciphers therefore they are very important for security of the ciphers. We must select S-Boxes carefully in order to be optimally resistant against known attacks.

Here, we gave a brief classification of the S-boxes criteria. We considered some important parameters of the S-boxes and presented the programs for their calculation which we wrote in C++ programming language. We also added some test results on different S-boxes with the running time, the size of the used RAM, and the size of the text file with the results. There are different methods and algorithms to calculate these parameters. Here, we mentioned some of them and explained what we have used.

We gave a concept for optimality of an S-box. Searching for optimal S-Boxes is a difficult task. There are many algorithms, using different representations of the optimal S-boxes, but still the problem remains.

In Section 5 we mentioned different techniques which can be used to obtain promising results. Parallel programing together with different methods and optimizing techniques can offer promising results in calculations of the properties and finding optimal S-boxes.

References:

[1] C. E. Shannon: “Communication Theory of Secrecy Systems.” Bell System Technical Journal, Vol 28, pp. 656–717, October 1949.

[2] G. Leander and A. Poschmann: “On the Classification of 4 Bit S-Boxes.” In C. Carlet and B. Sunar (Eds.): WAIFI 2007, LNCS 4547, pp. 159–176. Springer (2007).

[3] M. Matsui. *Linear cryptanalysis method for DES cipher*. In Advances in Cryptology – EUROCRYPT’93, vol. 765 of LNCS, pp.386-397, Springer Verlag, 1994.

[4] Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1990).

[5] M. Ullrich, C. De Cannière, S. Indesteege, Ö. Küçük, N. Mouha, and B. Preneel: “Finding Optimal Bitsliced Implementations of 4x4-bit S-Boxes.” SKEW 2011 Symmetric Key Encryption Workshop, Copenhagen, Denmark, 16-17 February 2011.

[6] J. Daemen and V. Rijmen. The Design of Rijndael. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[7] An Braeken (March 2006): Cryptographic Properties of Boolean Functions and S-Boxes, PhD Thesis, Katholieke Universiteit Leuven.

[8] Claude Carlet, **“**Vectorial Boolean Functions for Cryptography”, Chapter of the monography “Boolean Models and Methods in Mathematics, Computer Science, and Engineering”, Cambridge University Press, 2010.

[9] M.J. O. Saarinen: "Cryptographic Analysis of All 4x4 Bit SBoxes." In A. Miri, S. Vaudenay (Eds.): Selected Areas in Cryptography 18th International Workshop, SAC 2011. Toronto, ON, Canada, August 1112, 2011, Revised Selected Papers. LNCS 7118, pp. 118133. Springer (2012)